Novel simulation-based algorithms for optimal open-loop and closed-loop scheduling of deficit irrigation systems

N. Schütze, M. de Paly and U. Shamir

ABSTRACT

The scarcity of water compared with the abundance of land constitutes the main drawback within agricultural production. Besides the improvement of irrigation techniques a task of primary importance is solving the problem of intra-seasonal irrigation scheduling under limited seasonal water supply. An efficient scheduling algorithm has to take into account the crops' response to water stress at different stages throughout the growing season. Furthermore, for large-scale planning tools compact presentations of the relationship between irrigation practices and grain yield, such as crop water production functions, are often used which also rely on an optimal scheduling of the considered irrigation systems. In this study, two new optimization algorithms for single-crop intra-seasonal scheduling of deficit irrigation systems are introduced which are able to operate with general crop growth simulation models. First, a tailored evolutionary optimization technique (EA) searches for optimal schedules over a whole growing season within an open-loop optimization framework. Second, a neuro-dynamic programming technique (NDP) is used for determining optimal irrigation policy. In this paper, different management schemes are considered and crop-yield functions generated with both the EA and the NDP optimization algorithms compared.

Key words | closed-loop control, deficit irrigation, evolutionary algorithm, irrigation scheduling, neuro-dynamic programming, open-loop control

N. Schütze (corresponding author)
Institute of Hydrology and Meteorology,
Dresden University of Technology,
D-01062 Dresden,
Germany
E-mail: *ns1@rcs.urz.tu-dresden.de*

M. de Paly
Wilhelm-Schickard-Institute for Computer Science,
University of Tübingen,
Sand 1, D-72076 Tübingen,
Germany

U. Shamir
Technion–Israel Institute of Technology,
Haifa 32000,
Israel

NOTATION

B_n	elitist set of a generation
$\mathbf{c}$	center vector of RBF
d_i	date of irrigation event i
$d_{\min}$	minimum time between two irrigations
eps	termination criteria
j	number of growing period (1 = initial, 2 = crop development, 3 = mid-season, 4 = late season)
$K_{Y,j}$	crop sensitivity factor for period j
M	number of growing periods
n_j	index over stages belonging to the jth period
n_t	tournament size
n_{gen}	number of individuals in a population
$n_{\max}$	maximum number of iterations
N	number of stages in a growing season
p_{t}	take-over probability
p_{cr}	crossover probability
$\mathbf{s}$	pair of irrigation parameters for one irrigation event
$\mathbf{S}^{\pi}$	series of irrigation actions – schedule in NDP
$\mathbf{S}$	irrigation schedule for one growing season
U	cost-to-go function
$\tilde{U}$	approximated cost-to-go function
U^{*}	optimal cost-to-go function
v_i	water volume for irrigation event i
V_0	given water volume
V_i	remaining water volume until end of growing season
$V_{\min}$	minimum water volume per irrigation event
$V_{\max}$	maximum water volume per irrigation event
$\mathbf{W}$	weight matrix
$\mathbf{x}$	state vector
x	state variable
X	actual population of irrigation schedules
$\mathbf{X}$	set of all individuals of all generations

doi: 10.2166/hydro.2011.073

y_i daily contribution to yield response
Y relative yield
Y_a actual yield
$Y_{\max}$ maximum yield
$\mathbf{z}$ eligibility vector
δ_i temporal difference
λ discount parameter
π policy or decision function
π^* optimal policy
ϕ basis function
σ radius of an RBF
σ_{d} mutation rate for irrigation dates
σ_{v} mutation rate for water volumes
θ_r residual water content
θ_s saturated water content
ω weight

INTRODUCTION

The great challenge of the agricultural sector is to produce more food and/or more revenue from less water, which can be achieved by optimal irrigation management. A task of primary importance is the problem of intra-seasonal irrigation scheduling (i.e. when and how much to irrigate) under limited seasonal water supply. Here, a limited amount of water has to be distributed over a number of irrigations, taking into account the crop's response to water stress at different stages during the growing season. Dynamic programming (DP) has been extensively used for the optimization of closed-loop irrigation scheduling problems (Bras & Cordova 1981; Rao *et al*. 1988; Sunantara & Ramirez 1997; Prasad *et al*. 2006). An alternative approach to calculate optimal irrigation schedules is provided by open-loop scheduling techniques such as linear and nonlinear programming (Shang & Mao 2006; Gorantiwar *et al*. 2006).

Open-loop optimization is based on forecasts generated by simulation or analytic functions (Shani *et al*. 2004) of the water balance and crop production of an irrigation system for a whole growing period in advance. The open-loop irrigation scheduling problems can be formulated in two ways. The first way considers the water volume of each day of the growing season as a decision variable, resulting in a hard to solve nonlinear optimization problem (NLP) with a high number of decision variables. The other way significantly reduces the size of the search space by considering only actual irrigation events (i.e. dates and amounts) leading to a mixed integer nonlinear optimization problem (MINLP) with an *a priori* unknown number of decision variables. Therefore, recent studies simplify the optimization problem by fixing the irrigation dates (Loganathan & Elango 2004; Shang & Mao 2006) or the irrigation intervals (Montesinos *et al*. 2002; Gorantiwar *et al*. 2006; Brown *et al*. 2006). Beside these approaches heuristic optimization algorithms where used like Nelder–Mead simplex method (Shang & Mao 2006) or simulated annealing (Brown *et al*. 2006), which may fail in practice when local optimal solutions exist or when the number of decision variables becomes too large.

Alternatively, the problem can be solved by a closed-loop optimization strategy like DP, which is designed to obtain a lookup table containing optimal decisions for each possible state of the soil–vegetation–atmosphere system at each stage of the growing season. Depending on the definition of the state variables (e.g. soil moisture distribution and crop growth status) state updating based on real-time measurements can be used in order to adjust irrigation decisions and thus the optimal scenario of crop growth without recalculating the lookup table. The popularity and success of this technique can be attributed to the fact that nonlinear and stochastic features of scheduling problems can be handled by DP (Bertsekas 2000). However, it is well known that computational requirements of DP become overwhelming when the number of state and control variables is too large (Bellman & Dreyfus 1962). For this reason all the studies applying DP for optimal irrigation scheduling have their limitations because they use discrete representations for both state space and decision space. Bras & Cordova (1981) divided a growing season into 15 decision stages, which correspond to fixed irrigation intervals of 8 days. Five different irrigation policies (from irrigating up to field capacity down to not irrigating at all) were considered and a rough discretization of the state variables (soil moisture and available irrigation water) was used. Rao *et al*. (1988) employed DP for optimal water allocation over four growing stages combined with a heuristic method for the distribution of the allocated water in weekly irrigation intervals during each crop growth stage. A further development of the approach proposed by Sunantara & Ramirez (1997) avoids separating the optimization process into a DP part and a heuristic part. Daily irrigation decisions, however, would allow a more precise optimization of the amount and date of the irrigation events and the resulting lookup table would show its flexibility when constraints (like fixed irrigation intervals or fixed irrigation amounts) are present. Moreover, it is unfortunate that almost all the optimal scheduling procedures proposed so far rely solely upon water balance models although the process modeling of soil water transport offers a far more accurate representation of reality (Schmitz *et al.* 2007).

All attempts to use more comprehensive simulation models in irrigation scheduling employ trial-and-error methods, i.e., the generation and evaluation of a large set of arbitrary chosen scenarios (Raghuwanshi & Wallender 1997; Scheierling *et al.* 1997; Singh & Singh 1997; Shang *et al.* 2004). Raghuwanshi & Wallender (1997) constructed a seasonal furrow irrigation model (FIM) based on kinematic-wave hydraulics to minimize seasonal irrigation cost for a prescribed irrigation adequacy. This technique, however, comprises also some restrictions due to: (i) a constant irrigation interval and (ii) optimization by enumeration of all possible strategies. This severely limits the complexity of considered strategies in order to keep the 'optimization' computationally feasible. Singh & Singh (1997) calculated water management response indicators (WMRI) for different soil types which primarily prevent deep percolation. WMRI are based on a number of irrigation scenarios simulated by the water flow and transport model SWASALT. Also Scheierling *et al.* (1997) used a dynamic water flow model based on the Richards equation for evaluating a number of $2^9 = 512$ schedules (i.e. an enumeration of binary control vectors which represent a schedule of nine irrigation decisions). They found that crop yields vary enormously depending on the timing of irrigation. Considering the computational effort of this enumeration scheme makes it obvious that more efficient optimization methods are necessary. Shang *et al.* (2004) carried out nine simulations on the variation of soil moisture and irrigation scheduling and concluded that simulations of water dynamics under different irrigation conditions are essential for irrigation planning.

The objective of this study is to demonstrate the feasibility and effectiveness of a simulation–optimization strategy for open- and closed-loop optimization of irrigation scheduling which overcomes the above-mentioned restrictions. The simulation–optimization approach combines a broader range of simulation models with an optimization algorithm for solving deterministic and stochastic optimization problems (e.g. to handle the uncertainty to account for the impact of climate and soil variability is considered in optimal scheduling). In the context of a simulation-based optimization, a simulation model can be thought of as a function that turns input parameters into output performance measures that can only be evaluated by computer simulation (Gosavi 2003). As such, these functions are usually considered as a black box for the optimization algorithm. Evolutionary or genetic algorithms (EAs) are popular heuristic methods which are capable of achieving global or near-global optimal solutions to open-loop simulation-optimization problems. The significant advantage of the EA is that it can be directly linked with irrigation simulation models without requiring further model simplifications or the calculation of derivatives (Onwubolu & Babu 2004). In the field of irrigation, genetic algorithms were only applied to related problems to irrigation scheduling, for example optimal irrigation reservoir operation (Loganathan & Elango 2004; Wardlaw & Bhaktikul 2004; Kumar *et al.* 2006) or water delivery scheduling for an open-channel irrigation system (Nixon *et al.* 2001). This paper introduces a problem-specific EA which explicitly accounts for all possible constraints in intra-seasonal irrigation scheduling.

For solving dynamic simulation–optimization problems neuro-dynamic programming (NDP) is employed in this study. NDP belongs to the class of reinforcement learning methods, reducing the numerical complexity of standard DP. It avoids the exponential increase of computations through the use of parametric approximate representations of the cost-to-go function (Bertsekas 2000). Compared to the classical numerical solution approach for DP, which performs exhaustive sampling of the entire state space in solving the stage-wise optimization, these approaches sample only a small, crucial fraction of the state space and thus require less computations.

The remainder of this paper is organized as follows. In the methodology section, we review the new EA for intra-seasonal irrigation scheduling and the least-squares temporal difference (LSTD) algorithm for calculating the approximate cost-to-go function for the DP approach. In the Results section, a case study involving deficit irrigation of corn is presented to illustrate the new methods and we discuss the results, especially crop-yield functions generated using both the open- and closed-loop simulation-optimization. Finally, we offer some conclusions and suggestions for potential stochastic applications, especially for the NDP approach.

METHODOLOGY

When irrigation is constrained by limited water availability, a maximum crop yield is not achievable. With deficit irrigation, the plants are consciously under-supplied with water and a reduced crop yield is accepted as the penalty. However, each plant's level of water stress sensitivity fluctuates with respect to its different growth phases. For this reason, when laying down the irrigation schedules for an entire growth period, it is important to decide beforehand when the crop in a growth phase requires generous irrigation water volumes and, on the other hand, when smaller volumes will suffice. The objective of the simulation–optimization is to achieve maximum crop yield with a given, but limited water volume, which can be arbitrary distributed over an adequate number of irrigations. The impact of different irrigation schedules on crop yield is calculated by an irrigation model. For the sake of simplicity, an irrigation water balance model (Rao et al. 1988) is used in this study. The model computes the average volumetric water content θ_i in the soil at each stage which is determined by

$$\theta_i = \min\left(fc,\, \max\left(pwp,\, \frac{\theta_{i-1}D_{i-1} + v_i + P_i - \mathrm{AET}_i + \theta_0(D_i - D_{i-1})}{D_i}\right)\right) \quad (1)$$

where D_i is the actual rooting depth, P_i is the precipitation, v_i is the irrigation water volume, AET_i is the actual evapotranspiration (AET), fc is field capacity and pwp is the permanent wilting point of the considered soil. AET_i is computed as the sum of evaporation and transpiration by the plants and depends on the type of the crop and the potential evapotranspiration (PET_i). It is calculated according to

$$\mathrm{AET}_i = \begin{cases} \mathrm{PET}_i & \theta_i D_i \geq 1 - p(\mathrm{PET}_i)\,(fc - pwp)D_i \\ \dfrac{\theta_1 D_i \mathrm{PET}_i}{(1 - p(\mathrm{PET}_i))(fc - pwp)D_i} & \theta_i D_i < (1 - p(\mathrm{PET}_i))\,(fc - pwp)D_i \end{cases} \quad (2)$$

where $p(\mathrm{PET}_i)$ is the crop-dependent soil water depletion factor. The actual yield Y_a is computed from the resulting values of AET_i according to the multiplicative FAO-33 crop yield response model (Doorenbos & Kassam 1979) which is

$$Y = \frac{Y_a}{Y_{\max}} = \prod_{j=1}^{M}\left(1 - K_{Y,j}\left(1 - \frac{\sum_{i=n_{j-1}+1}^{n_j} \mathrm{AET}_i}{\sum_{i=n_{j-1}+1}^{n_j} \mathrm{PET}_i}\right)\right) \quad (3)$$

where Y is the relative yield and $Y_{\max}$ the maximum yield.

Formulation of the open-loop scheduling problem

The objective of open-loop optimization is to achieve maximum crop yield Y with a given, but limited, water volume V_0. V_0 has to be distributed over the growing season, where the time and the quantity of each irrigation have to be determined. The impact of an irrigation schedule on the crop yield is calculated by an arbitrary seasonal irrigation water balance model, e.g., Rao et al. (1988), or a more comprehensive agricultural production model. The global optimization problem can then be formulated as an MINLP with continuous and discrete decision variables as follows:

$$Y^* = \max Y(\mathbf{S}): \quad \mathbf{S} = \{\mathbf{s}_i\}_{i=1,\ldots,n} = \{(d_1, v_1), \ldots, (d_i, v_i), \ldots, (d_n, v_n)\}\, n,\, d_i \in \mathbf{N} \cdot v_i \in \mathbf{R} \quad (4)$$

with the optimal solution for maximizing the yield Y:

$$S^* = \arg\max Y(\mathbf{S}) = \arg\max Y(\{(d_i, v_i)\})\, i = 1, \ldots, n \quad (5)$$

where $\mathbf{S}$ is the schedule for the whole growing season, consisting of $i = 1, \ldots, n$ irrigation events $\mathbf{s}_i$ each defined by the date d_i and the irrigation depth v_i. The number n of irrigation events $\mathbf{s}_i$ is not fixed *a priori* and is a decision variable itself. The set of feasible schedules is determined by the three following constraints:

$$\sum_{i=1}^{n} v_i \leq V_0 \tag{6}$$

$$|d_i - d_j| \geq d_{\min} \quad \forall d_i, d_j \in \mathbf{S}; \quad i \neq j \tag{7}$$

$$V_{\max} \geq v_i \geq V_{\min} \tag{8}$$

i.e.,

- Equation (6) limits the sum of the irrigation depth for the growth period which must not exceed the given water volume V_0,
- Equation (7) sets a minimal time between two irrigations which must not fall below $d_{\min}$,
- Equation (8) provides bounds for each single irrigation depth which must be within the prescribed range [$V_{\min}$, $V_{\max}$].

Solving the open-loop optimization problem with a new EA

The EA begins with a set of solutions, called population, which, in our case, is a random set of schedules. Every member of the set has a fitness value assigned that is directly related to the objective function – its crop yield. In sequential steps, the population of schedules is modified by applying the four operators: selection, crossover, mutation and reconstruction. These operators mimic their counterparts from the natural evolution process. Selection chooses individuals according to their fitness from the previous generation, which are the base for the individuals of the new generation. The offspring are either generated by crossover which combines two individuals into a new one or by mutation which randomly changes a single individual. The details of the algorithm are presented in Algorithm 1 and Algorithms 3–6 in the appendix. The main features are as follows.

A population X^n is a set of n_{gen} individuals, i.e., irrigation schedules $\mathbf{S}$. Each irrigation schedule consists of a set of pairs $\mathbf{s}_i = (d_i, V_i)$, where each pair contains the parameters of an individual irrigation event. The entire set $\mathbf{X}$ – i.e., all those individuals which are created during the optimization – can be formed by bringing together all the individuals of all generations:

$$\begin{aligned}\mathbf{X} &= \{X^n\}_{n=1,\ldots,n_{\max}} = \{\{\mathbf{S}_j\}^n_{j=1,\ldots,n_{\text{gen}}}\}_{n=1,\ldots,n_{\max}} \\ &= \{\{\{(d_i, v_i)\}^j_{i=1,\ldots,n_j}\}^n_{j=1,\ldots,n_{\text{gen}}}\}_{n=1,\ldots,n_{\max}}.\end{aligned} \tag{9}$$

The convergence and outcome of the EA are determined by the following parameters: the maximum number of function

Algorithm 1 Main loop of the EA.

▷ assign parameters n_{max}, n_{gen}, n_t, eps, σ_d, σ_V, p_t, p_{cr}, d_{min}, V_0, $n = 0$
▷ initialize randomly $X^0 = \{\mathbf{S}_j\}^0_{j=1\ldots n_{gen}}$
▷ construct feasible schedules **foreach** $\mathbf{S}_j \in X^0$ by *reconstruction*($\mathbf{S}_j$, d_{min}, V_0) – (see Alg.6)
▷ calculate $\mathbf{Y}(\mathbf{S}_j)$ for all $j = 1 \ldots n_{gen}$
while $(n < n_{max})$ or $(\Delta_{n-(n-1)}\,\mathbf{Y}(X) \leq eps)$
  $B_n = \emptyset$, $X^{n+1} = \emptyset$
  for $(j = 1; j \leq n_{gen}; j++)$
    ▷ calculate B_n and $\mathbf{S}_j'$ by *tournament-selection*($\mathbf{S}_j$, X^n, B_n, n_t) – (see Alg.3)
    if $(\mathbf{S}_j \notin B_n)$
      if $(p_{cr}(\mathbf{S}_j))$
        ▷ calculate offspring $\mathbf{S}_j$ by crossover ($\mathbf{S}_j$, $\mathbf{S}_j'$, p_t) – (see Alg.4)
      endif
      ▷ calculate $\mathbf{S}_j$ by *mutation*($\mathbf{S}_j$, σ_d, σ_V) – (see Alg.5)
      ▷ construct a feasible schedule $\mathbf{S}_j$ by *reconstruction*($\mathbf{S}_j$, d_{min}, V_0) – (see Alg.6)
      ▷ calculate $\mathbf{Y}(\mathbf{S}_j)$
    endif
    ▷ $X^{(n+1)} = \{X^{(n+1)}, \mathbf{S}_j\}$
  endfor
  ▷ $n++$
endwhile

evaluations $n_{\max}$, the stopping criteria *eps* – the difference of the maximum objective function values in the population between two consecutive generations – the mutation rate for irrigation dates σ_{d} and volumes σ_{v}, the takeover probability p_{t} and the crossover probability p_{cr}.

The structure of the EA shown in Algorithm 1 deviates in certain aspects from the standard operators of EAs – selection, crossover and mutation. First, the deviations include a change in the order in which the individual operators are activated. During each generation step the selection is the first operation to be carried out, instead of at the end. Second, an additional reconstruction step rebuilds the created children in order to guarantee feasible solutions which are in compliance with the constraints. For doing this we used *a priori* knowledge about irrigation scheduling, e.g., that it is better to irrigate for future crop requirements in advance than to irrigate too late. The implementation of the operators is explained subsequently (for details see Algorithms 3–6 in the appendix).

Selection

To determine the parents of the next generation we employ an elitistic tournament selection. This means the algorithm iterates over all individuals of the current generation. Each individual is set once as one participant of a tournament with a set of n_{t-1} randomly chosen competitors from the same generation, where n_t is the tournament size. If the set individual wins – i.e., it has the better objective function value – it is retained in the elitist set of the generation B_n and goes unchanged in the next generation. Otherwise a new individual is generated from the best competitor using the crossover, mutation, and reconstruction operators.

Crossover

The number of irrigation events can differ between the two parent individuals. Thus, it is not possible to use one of the standard crossover operators. Instead, the crossover operator must be altered to suit the structure of the data. Because plant water uptake is time-dependent, with respect to crossover it makes sense to preserve the relationship between the irrigation time and volume of the schedules of the two parents. This can be achieved by creating the offspring individual out of a selection of irrigation events (pairs $\mathbf{s}_i$), which themselves are chosen from the combined total of the parents' own irrigation schedules. Thus, in implementation of the crossover operator each irrigation event from the set union of the parents' irrigation schedules is selected with a certain probability p_{t} and placed into the offspring schedule.

Mutation

For all the irrigation times d_i and irrigation volumes v_i of an irrigation schedule, mutation is implemented by adding a normally distributed random value, which has to be generated for each variable to be mutated. Different crops react differently to changes made to the irrigation timing and/or to the water volumes. For this reason, we distinguish between the variances for the mutation of the irrigation times σ_{d} and the variances for the mutation of the irrigation volumes σ_{v} to control the mutation.

Reconstruction

Schedules of the new population are reorganized in the following manner: two water applications spaced by an interval smaller than the given minimal irrigation interval are combined into one water application. The water volumes of the two are added and the irrigation time of the earlier event is selected for the combined event. All the other water applications remain in the schedule without change. Thereafter, the amount of each water application is normalized to meet the total available water volume with the sum of the individual irrigation water volumes. Once these steps have been applied to the whole population, irrigation simulations are performed with all the new individuals (schedules).

After this step one generation of an EA is completed. The algorithm iterates until a certain desired degree of convergence is reached.

With respect to the fact that there are numerous examples of general evolutionary optimization procedures in the literature, it is worth noting that optimal irrigation scheduling is a challenging open-loop optimization problem. In a recent study (de Paly & Zell 2009) we compared the performance of the new developed algorithm with six state-of-the-art general EAs, namely real-valued genetic algorithm, particle swarm optimization, differential evolution, evolution strategy, covariance matrix adaptation evolution strategy and shuffled complex evolution. Each algorithm had the task of minimizing the yield loss while distributing a given amount of water V_0 over the entire growth period of 130 day, with a possible irrigation at each day based on the same irrigation model as used in this paper (see Equation (1)). The resulting constrained high-dimensional NLP showed to be hard to solve for the general EA, which do not employ problem-specific operators. From Figure 1, it can be seen that no general EA is able to find the global optimal schedule within the given maximum number of 5,000 function evaluations.

EAs usually require many function evaluations for convergence, making them computationally intensive. The presented EA reduces the computational effort by restricting the number of individuals, which have to be evaluated by simulations, to feasible solutions. In addition, the overall time necessary for one optimization run can be reduced through extensive parallel processing of objective function evolution for all individuals of one generation at once. At present, interfaces to APSIM (Keating *et al.* 2003), DSSAT (Jones *et al.* 2003), PILOTE (Kholedian *et al.* 2009) and DAISY (Abrahamsen & Hansen 2000) crop growth models as well as the FAO-33 yield response model are implemented.

Formulation of the closed-loop optimization problem

As a general framework for solving the closed-loop optimal scheduling problem we used a dynamic program, consisting

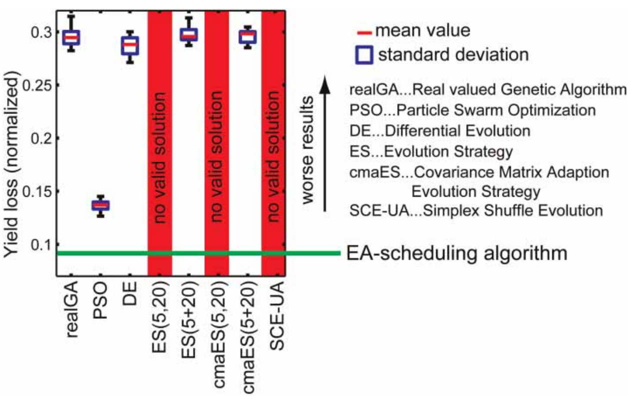

Figure 1 | Performance of six general evolutionary and the new tailor-made scheduling EA from de Paly & Zell (2009). For a detailed description of the general evolutionary algorithms see Streichert & Ulmer (2005).

of state variables, decision variables, a transition function, a contribution or cost function, an objective function and a decision function. For the irrigation scheduling problem the state variables are the average soil water content θ_i which represents the amount of water in the soil water reservoir and V_i the volume of water which is available for irrigation from stage i until the end of the time horizon. The decision variable at each stage is the depth of irrigation water v_i which can be applied. Daily decisions are provided by the decision function or operation policy π specified in the form $v_i = \pi(V_i, \theta_i)$. The transition function which describes the dynamics of the irrigation system, i.e., the irrigation model, is defined as in Equations (1) and (2). The objective function of the dynamic problem for a limited water supply can then be formulated as

$$\begin{aligned} Y^* &= \max_{\text{all}\, \mathbf{S}^\pi}\left(\sum_{i=1}^{N} y_i(V_i, \theta_i, v_i)\right) \text{ with} \\ \mathbf{S}^\pi &= \{v_1, \ldots, v_i, \ldots, v_N\} \text{ subject to } \sum_{i-1}^{N} v_i \le V_0 \end{aligned} \tag{10}$$

where $\mathbf{S}^\pi$ is a series of actions, i.e., an irrigation schedule with the daily irrigation depth v_i at stage i. The reward y_i is the daily contribution to the crop yield response, which is determined by an additive formulation (see Equation (11)) derived from the multiplicative FAO-33 crop yield response model given in Equation (3). y_i can be interpreted as the contribution to the reduction of crop yield response related to potential yield as a result of actual water stress (see Equation (12)):

$$\begin{aligned} Y = \frac{Y_a}{Y_{\max}} = 1 + \sum_{j=1}^{M}\left(\left(\prod_{k=0}^{j-1} Y_k\right) K_{Y,j}\right. \\ \left. \times \sum_{i=n_{j-1}+1}^{n_j}\left(\frac{\mathrm{AET}_i}{\sum_{l=n_{j-1}+1}^{n_j}\mathrm{PET}_l} - \frac{1}{n_j - n_{j-1}}\right)\right) \end{aligned} \tag{11}$$

$$y_i = \left(\prod_{k=0}^{j-1} Y_k\right) K_{Y,j}\left(\frac{\mathrm{AET}_i}{\sum_{l=n_{j-1}+1}^{n_j}\mathrm{PET}_l} - \frac{1}{n_j - n_{j-1}}\right) \tag{12}$$

where Y_k is the cumulative yield reduction according to the crop sensitivity factor $K_{Y,j}$ for the past growing periods

$k = 1, \ldots, j-1$. j is the actual growing period with the crop sensitivity factor $K_{Y,j}$ which starts at decision stage $n_{j-1}+1$ and ends at stage n_j. The other variables related to the FAO-33 crop yield response model are the number of growing periods M, PET, AET and relative yield Y. The AET depends on the water balance model used which is defined in Equations (1) and (2) and thus y_i depends on the state variables θ_i and V_i and the decision variable v_i. Through the introduction of the optimal cost-to-go function U^* for each state (V_i, θ_i) given by the recursive equation of the DP:

$$U_i^*(V_i, \theta_i) = y_i(V_i, \theta_i, v_i) + \max_{v_i \in [0, \min(V_i, V_{\max})]} [U_{i+1}(V_i - v_i, \theta_{i+1})] \quad \text{for} \quad i = 1, \ldots, N-1 \tag{13}$$

and

$$U_N^*(V_N, \theta_n) = y_N(V_N, \theta_N) \tag{14}$$

we are able to find the optimal policy or decision function π^* calculating

$$\pi^*(V_i, \theta_i) = \arg \max_{\mathrm{v_i} \in [0, \min(V_i, V_{\max})]} (y_i(V_i, \theta_i, v_i) + [U_{i+1}(V_i - v_i, \theta_{i+1})]). \tag{15}$$

The optimal cost-to-go can be interpreted as the minimum reduction in yield for the time period that remains after a time i. To solve the optimality Equation (13) by DP, a sequential calculation of U_i^* is performed for all stages and all states at each stage by backtracking starting from the terminal stage N.

Solving the closed-loop optimization problem with NDP

Classical DP is based on the premise that the number of states $\mathbf{x}$ of a system is finite. This is not the case if we apply irrigation simulation models which use continuous variables ($x_1, \ldots, x_n$). The simulation-based approach of DP used here is neuro-dynamic programming (NDP), which approximates the cost-to-go function $U^*(\mathbf{x})$ by an approximation function $\widetilde{U}(\mathbf{x}, \mathbf{W})$ in an iterative loop. NDP uses linear basis function approximators (Taylor series, tile coding or radial basis function) or nonlinear universal approximators like multilayer perceptron to learn the cost-to-go function $\widetilde{U}$ (Sutton & Barto 1998). In this study we employed a linear approximation approach where the cost-to-go function is given by the linear combination of l basis functions ϕ_k:

$$\widetilde{U}(\mathbf{x}, \mathbf{W}) = \sum_{k=1}^{l} w_k \phi_k(\mathbf{x}) \tag{16}$$

with the parameter vector w_k and a radial basis function (RBF) as the choice of ϕ_k:

$$\phi_k(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_k\|^2}{2o^2}\right) \tag{17}$$

where σ is a suitable chosen radius and $\mathbf{c}_k$ are the centers of the l basis functions.

The weight matrix $\mathbf{W}$ has to be determined by some form of optimization, e.g., by using a least-squares framework, minimizing the error of the temporal differences (TDs):

$$\delta_i = \widetilde{U}_i(\mathbf{x}_i, \mathbf{W}) - (y_i + \widetilde{U}_{i+1}(\mathbf{x}_{i+1}, \mathbf{W})) \tag{18}$$

From Equation (19), which is related to the Bellmann Equation (13), it can be seen that TDs are the errors in the estimates of the approximated cost-to-go function $\widetilde{U}(\mathbf{x}, \mathbf{W})$ compared to the true reward y_i between two temporally successive predictions of the cost-to-go in an irrigation scenario. TDs δ_i would be equal to zero in the ideal case for all simulated states of the irrigation system for all irrigation scenarios if $\widetilde{U}(\mathbf{x}, \mathbf{W})$ would be equal to $U^*(\mathbf{x})$.

In this study, the cost-to-go approximation function is constructed by least-squares temporal differences policy evaluation LSTD(λ) (Boyan 2002) and ε-greedy policy improvement, which finds a new policy by maximizing the actual cost-to-go function in the space of feasible policies (Sutton & Barto 1998). For obtaining the approximation function $\widetilde{U}(\mathbf{x}, \mathbf{W})$ the policy iteration algorithm alternates between approximating *policy evaluation* steps and *policy improvement* steps (see Figure 2).

Before we describe the algorithm in brief we need to introduce the time t as a state variable. Since the application of the policy iteration algorithm requires a stationary policy, we define $v_i = \pi(\mathbf{x}) \equiv v_i = \pi(V_i, \theta_i, t_i)$. This is a precondition

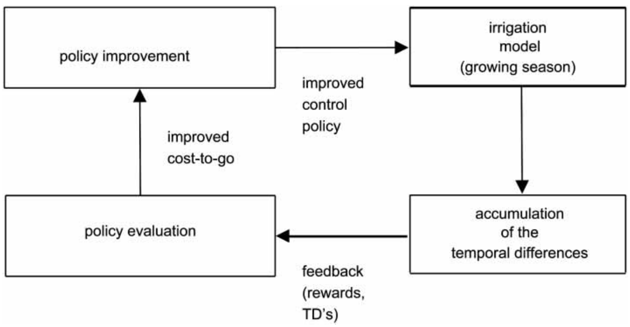

Figure 2 | Basic structure of the policy iteration algorithm.

for the application of the policy iteration algorithm. The policy iteration algorithm (see Algorithm 2 and Figure 2) contains the following procedures:

Simulation

The irrigation model simulates a scenario (trajectory) with the actual policy $\mathbf{S}_n^\pi$ and calculates the rewards $y_i(\mathbf{x}_i, \tilde{v}_i)$ for all the states that are on the trajectory.

Accumulation of the TDs

During the simulation on each state transition the TDs among all RBF ϕ are updated in $\mathbf{A}$ and $\mathbf{b}$ according to their respective eligibilities z_i. The eligibility vector can be seen as an algebraic trick by which TD propagates rewards backwards over the current scenario without having to remember the scenario explicitly. Thus, each RBF's eligibility at time i depends on the scenario's history. λ controls how the TD errors between successive predictions are passed back in time. If λ is set to 0, the error signal only propagates to the previous state. If it is set to 1, all previous states are affected by an exponentially decaying amount.

Policy evaluation

Updates of the approximation function $\widetilde{U}$ are carried out offline, i.e., the weights $\mathbf{W}$ of $\widetilde{U}$ are modified only at the end of each scenario by solving the linear least-squares problem $\mathbf{W} = \arg\min \| \mathbf{AW}\text{-}\mathbf{b} \|^2$ using the pseudoinverse of $\mathbf{A}$.

Policy improvement

The exploration policy uses an ε-greedy policy: The greedy action $\tilde{v}_i$ (i.e. the one for which the sum of the reward y_i and the successor states estimated cost-to-go $\widetilde{U}$ is the maximum) is chosen with probability $1 - \varepsilon$. In the other cases a random action is drawn from a uniform distribution over the range zero to the remaining water volume V_i. The value of ε is reduced during learning, until the policy

Algorithm 2 Approximate policy iteration using LSTD(λ)-policy evaluation.

▷ assign parameters n_{max}, eps, λ, σ, $\{\mathbf{c}_k\}$, d_{min}, V_0, V_{min}, V_{max}, $n = 0$

▷ initialize $\mathbf{S}^\pi$ using a random policy $\mathbf{S}_1^\pi = \{\text{rand}(\tilde{v}_i)\}$

while ($n < n_{max}$) or ($\mathbf{S}_{n-1}^\pi \equiv \mathbf{S}_n^\pi$)

▷ $\mathbf{A} = 0$; $\mathbf{b} = 0$; $n = n + 1$;

for ($i = 1$; $i < N$; i++)

▷ calculate $y_i(\mathbf{x}_i, \tilde{v}_i)$ and $\mathbf{x}_{i+1}$ using simulation

▷ accumulate the temporal differences

$\mathbf{A} = \mathbf{A} + \mathbf{z}_i\,(\phi(\mathbf{x}_i) - \phi(\mathbf{x}_{i+1}))^T$

$\mathbf{b} = \mathbf{b} + \mathbf{z}_i y_i$

$\mathbf{z}_{i+1} = \lambda \mathbf{z}_i + \phi(\mathbf{x}_{i+1})$

endfor

▷ evaluate the approximate cost-to-go $\tilde{U}(\mathbf{x}, \mathrm{W})$ using pseudoinverse of $\mathbf{A}$

$\mathbf{W} = \mathbf{A}^{-1}\mathbf{b}$

▷ improve policy

$$\mathbf{S}_{n+1}^\pi = \{\tilde{v}_i\} \text{ with } \begin{cases} p(1-\epsilon) \rightarrow \tilde{v}_i = \underset{v_i \in [0, \min(V_i, V_{max})]}{\arg\max} \left[y_i(V_i, \theta_i, t_i) + \tilde{U}(V_i - v_i, \theta_{i+1}, t_i, \mathbf{w}) \right] \\ else \quad \rightarrow \tilde{v}_i = \text{rand}([0, V_i]) \end{cases}$$

subject to $\text{abs}(d_i - d_j) \geq d_{min}$ and $V_{max} \geq \tilde{v}_i \geq V_{min}$ for all $i = 1 \ldots N - 1$

endwhile

improvement step converges to an entirely greedy behavior. For obtaining the greedy action $\tilde{v}_i$ a line search method is employed.

In our study we used a random generated initial policy, i.e., a random initial irrigation schedule that was appropriate for the considered problem. The policy iteration algorithm continues until suboptimal stable policies are achieved, which is also reflected by good returns from the approximate cost-to-go function. Good estimates of the initial policy can be used to accelerate the convergence of $\widetilde{U}$ and thus speed up the convergence of the entire algorithm.

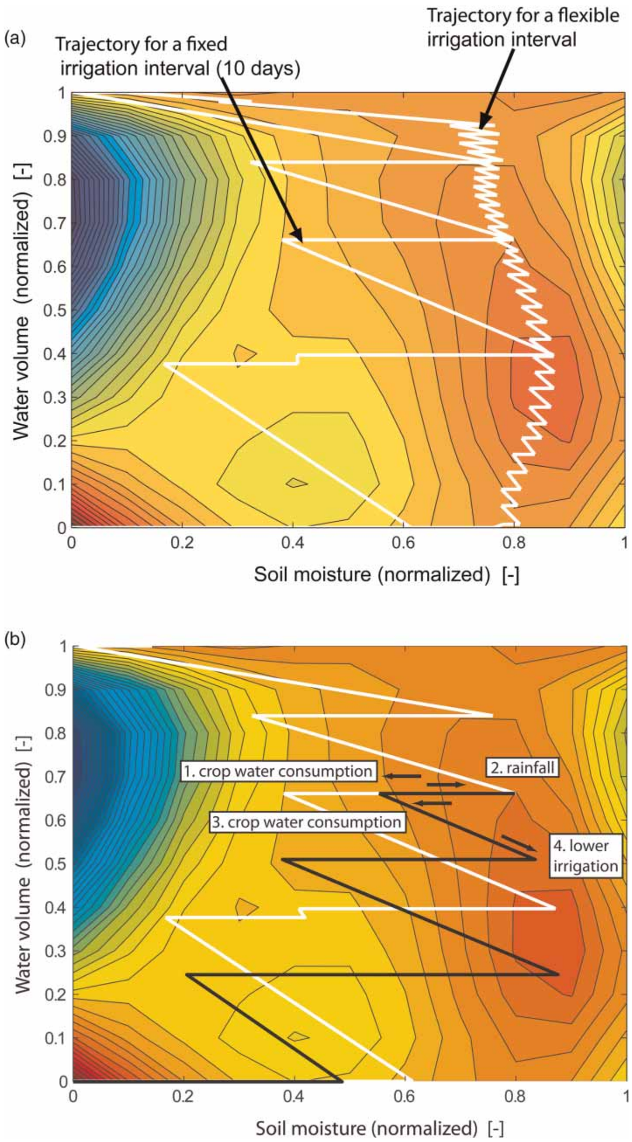

Figure 3 | Approximated cost-to-go function in two dimensions with trajectories (schedules) for different management constraints and climate conditions. (Colors indicate maximum expected relative yield (blue/black: low; red/white: high; water volume is normalized with respect to V_0, soil moisture is normalized with respect to θ_s).

In order to give an idea of the shape of the cost-to-go function a simplified two-dimensional example is shown in Figure 3. From the sample trajectories it can be seen how the dynamic control of irrigation works. Two kinds of changes in the state space are apparent. First, horizontal movements are changes in the mean soil moisture either caused by crop water consumption or in the other direction (see Figure 3(b)) by rainfall. Second, a slant direction in the movement corresponds to irrigation events when the available water volume is reduced and the mean soil moisture increases (see Figure 3(a)). An optimal irrigation scenario uses the path along the highest values of the cost-to-go function it can reach.

For the sake of simplicity basic state variables and decision variables are used in this paper which need to be extended when dealing with more complex transition models, e.g. more complex SVAT (soil–vegetation–atmosphere transfer) models which also include 1D or 2D water transport. This is necessary if a more precise control of the distribution of water around the emitter of high performance irrigation systems, such as surface or subsurface drip irrigation systems, is of importance. Then classical DP would be hard to solve and even for NDP a low-dimensional representation of the spatial distribution of water in the soil is required (Hinnell et al. 2010). In addition, the dimensionality of the decision space increases considerably if management of fertilization and leaching is considered at the same time. In all those cases neural-dynamic programming may be the only approach that can be used.

APPLICATION TO INTRA-SEASONAL SCHEDULING IN DEFICIT IRRIGATION

We compared three management schemes in order to analyze the performance of the new scheduling algorithms. First, a fully flexible scheme where no dates and no volumes were fixed (referred to as 'flexible') is used. Second, a simplified scheduling problem is solved, where the possible dates of the irrigation events were fixed at multiples of 10 days (referred to as 'fixedD'). The third, and most inflexible, management scheme has the limitations of the second one and, in addition, only fixed irrigation volumes ($v_i = 50$ mm) were allowed (referred to as 'fixedDV').

The irrigation scenario

In a real-case application a limited amount of 1–600 mm water had to be distributed with irrigation schedules optimized for maximum crop yield. Detailed and mostly unpublished data of field experiments in Lavalette (France) regarding volumetric soil moisture content, evapotranspiration and other aspects of the experiments were kindly provided by Mailhol (2005) from CEMAGREF (France). In our study, the simulations were carried out by a water balance model (Rao *et al*. 1988) based on these experiments. In the irrigation scenario corn is grown over a growing period of 132 days starting from 26 May 1999. The irrigated field is a plot of silty loam, characterized by a saturated soil moisture $\theta_s = 0.41$, a residual soil moisture $\theta_r = 0.05$ and field capacity at $fc = 0.4$. To get a picture of the meteorological situation the development of the PET is shown in Figure 4. Values for corn for the development of the root zone, crop sensitivity factors K_y and soil water depletion factor p were taken from Doorenbos & Kassam (1979).

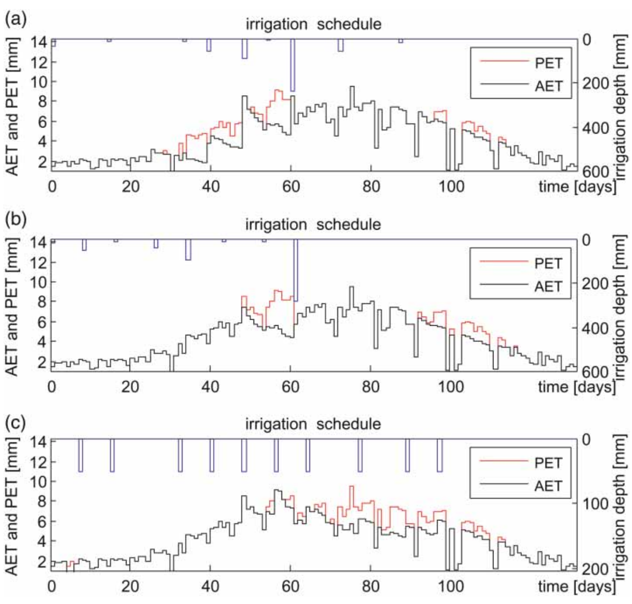

Figure 4 | Optimal schedules with a given water volume of 500 mm. Development of evapotranspiration for EA (a), NDP (flexible) (b) and NDP (fixedDV) (c).

The set-up of the EA

The basic parameters required by the EA are obtained by trial and error. In this study we used a population size of 50 schedules. The crossover probability and the take-over probability were $p_{\mathrm{cr}} = 0.33$ and $p_{\mathrm{t}} = 0.95$, respectively. The variances for irrigation dates were $\sigma_{\mathrm{d}} = 1.5$ d and for volumes were $\sigma_{\mathrm{v}} = 0.5$ mm. The length of an entire optimization run was limited to a maximum of 25 generations. In selection we applied the prescribed elitism procedure with a tournament size of $n_t = 4$. The minimal irrigation interval was set to 1 d for solving the open-loop optimization problem using the 'flexible' irrigation scheme. In order to generate the entire crop production function 61 optimization runs were carried out with the EA based on varying available irrigation water volumes $V_0 = \{0 \text{ mm}, \ldots, 10 \text{ mm}, \ldots, 600 \text{ mm}\}$.

The set-up of the NDP algorithm

In NDP the accuracy of the approximate cost-to-go function mainly depends on the number and the parameters of the chosen basis function, namely the radius of the Gaussian σ and their distribution in the state space. We fixed $\sigma = 0.1$ and considered only a variation of the number of RBF assuming always an uniformly spaced distribution of the RBF centers. Based on preliminary experiments the amount of RBF was fixed at $6 \times 6 \times 11$ according to the dimensions of the state space which was an acceptable trade-off between accuracy and speed of training of the approximator $\widetilde{U}$ $((V_i\}, \theta_i, t_i), \mathbf{W})$. The parameter λ in policy evaluation was set to 1 which leads to a supervised linear regression on the data of the simulated irrigation scenarios, i.e., the relationship of the simulated states and crop returns. For the policy improvement we started with an initial value $\varepsilon = 1$ which was gradually decreased with increasing number of training steps n as in $\varepsilon = \exp(-10n/n_{\max})$. In the case of NDP only a single application of the policy iteration algorithm was necessary to generate a universal approximate cost-to-go function which allowed us to perform all the optimization runs for each of the prescribed management scheme.

RESULTS

To examine the performance of the different optimization strategy consider Figure 4 which depicts the optimal irrigation schedule for a water volume of 500 mm and the corresponding development of PET and AET over the growing season for EA (flexible), NDP (flexible) and NDP (fixed DV), respectively. Figure 5 shows the development of the various parameters over the growing season: Figure 5(a) soil moisture and allowable depletion for the EA (flexible) case and Figure 5(b) crop sensitivity factor for corn in four periods of the growing season (0–25, 25–65, 65–105 and 105–132 days). The normalized crop production functions, which were generated under the 'flexible' scheme by the EA and under all schemes generated by NDP using the approximate cost-to-go, are presented in Figure 6.

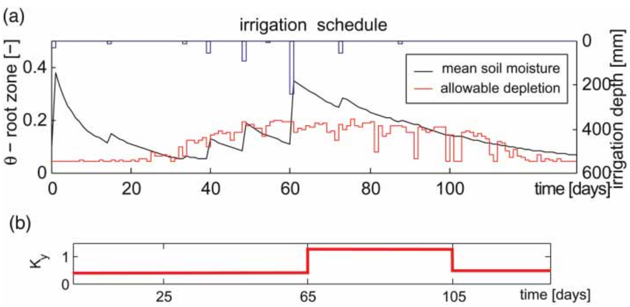

Figure 5 | Development of the cropping system with a given water volume of 500 mm. Development of the soil moisture for EA (a) and crop sensitivity factor K_y for corn (b).

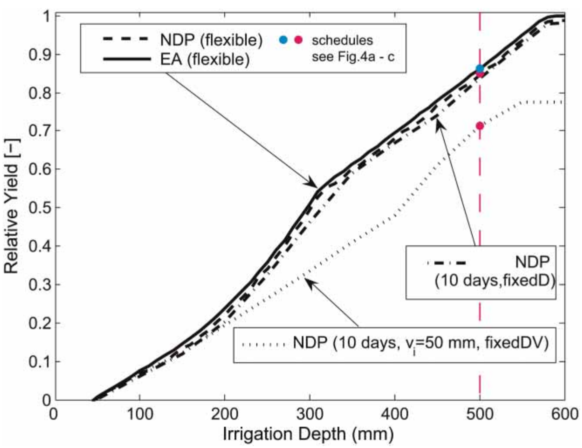

Figure 6 | Normalized seasonal crop production function.

From Figure 6, it can be seen that the EA achieved the best schedules, i.e., the highest yields for a given amount of water. The crop production function under the 'flexible' scheme is nonlinear in two ranges. The first range is in the vicinity of the point where all crop water requirements during a growing season are satisfied. The second range is between 200 and 300 mm of available water. At a water volume of 300 mm the crop water requirements of the third growth period, which has the highest stress sensitivity, are fully satisfied (which is $K_y = 1.3$ compared to $K_y = 0.4$ and 0.5 in the other crop growth periods – see Figure 5(b)). The nonlinearity is due to a side effect caused by a more and more adequate irrigation of the third period. The last growing period with a lower K_y and a higher allowable depletion (see Figure 5(a)) benefits disproportionately from the water which is stored in the soil at the time of transition from the third period to the fourth.

The results provided by an optimization using the approximate cost-to-go generated by the NDP algorithm and employing the 'flexible' scheme in the application are directly comparable to those determined by the EA. Slight variations can be observed which result in marginally modified schedules (see Figure 4(a,b)). The deviations of the NDP method are mainly caused by the approximation error which could be reduced by an increased number of RBFs. The crop production function under the 'fixedDV' scheme, which uses the same approximate cost-to-go function, shows a significant yield reduction caused by the limitations of this management scheme. An exception can be seen in the lower part of the crop production function for water volumes below 200 mm. The second nonlinearity range moved from 300 to 400 mm. This can be explained by the inflexibility of the management scheme, which does not always allow us to irrigate in an adequate way when the water stress sensitivity of corn is high. From a water volume of 550 mm onwards, there is no further improvement in the yield if more water is applied. This implies that all the additional water is percolating because field capacity was already been achieved in all days when irrigation is possible. The 'fixedD' scheme achieves better yields than the 'fixedDV' scheme, which are almost similar to the 'flexible' scheme. Some substantial deviations can be observed in the range between 200 mm and 400 mm where exact timing of the irrigations events is necessary in order to meet the crop water requirements of the most sensitive growth period.

Figure 4 shows optimal schedules generated by the EA and the NDP algorithm using the 'flexible scheme' for a given water volume of 500 mm. The development of the mean soil moisture in the soil (Figure 5(a)) relates to the soil water stored in the root zone whose depth increases linearly from 0.1 m to 1.2 m up to the 78th day and then remains constant. Figure 5(a) also shows the lower limit of the soil moisture where no reduction of the AET occurs. Figure 4(a,b) show that at the start of the third growth period an irrigation with a large amount of water is necessary, in order to accommodate (1) for the high water stress sensitivity and (2) for the low allowable depletion depth. As can also be seen, the schedule generated by the NDP algorithm tends to distribute more water in the first part of the growing season than the EA-generated one does.

This leads to a larger reduction of the AET in the last crop growth period resulting in diminished yields caused by the slightly higher value of $K_y = 0.5$ in the fourth growth period compared to $K_y = 0.4$ in the first and second growth period. The schedule generated by the application of the approximate cost-to-go employing the 'fixedDV' scheme shows a high density of irrigations before and shortly after the transition to the third crop growth period. This schedule can only partly account for the specific crop water requirements and leads to losses due to percolation (not shown in the graph).

We also investigated the computational efficiency of the EA and the NDP algorithm on a Pentium PC (2.8 GHz). In the EA case one optimization run needed less than a minute (convergence after a maximum of 1,000 function evaluations). But it has to be taken into account that one optimization run is necessary for each point of the crop production function. The computational effort of the NDP algorithm depends on various parameters. The LSTD algorithm for the policy evaluation has a cost of $O(N^2)$ for the accumulation of the TDs and $O(N^3)$ for the matrix inversion, where N is the number of the RBFs used. The line search in the policy improvement step required an average of 10 function evaluations but the computational costs increase only linearly with the number of iterations. The NDP algorithm converged after a maximum of 2,000 LSTD iterations and is able to provide a set of crop production functions for a specific site. Overall, the time for learning or approximating the cost-to-go function was around 10 h and the application time needed less than a second. However, the NDP methodology offers an improvement of the performance, taking into account that with the approximate cost-to-go function different tasks (different management schemes, different given amounts of volume, etc.) can be performed with a single (expensive) approximation step.

CONCLUSIONS AND FUTURE WORK

We presented two new optimization algorithms for simulation-optimization of scheduling under deficit irrigation throughout a whole growing season. If an open-loop optimization strategy is adopted, the tailor-made EA can be coupled with any irrigation model. In this case the model used for optimization has to be as accurate as possible and information about the future development of the climate variables is necessary or has to be provided by a framework for generating (stochastic) climate scenarios. With these preconditions the EA can provide optimal schedules which achieve maximum yield for a given amount of water within a reasonable computational time. The tailor-made EA is proven to be highly reliable compared to the Nelder–Mead simplex algorithm, simulated annealing and most recent general evolutionary optimization approaches. Therefore, the EA is the algorithm of choice if there is no lack of information and if the management scheme has no limitation (schemes such as the 'fixedD' and the 'fixedDV' schemes were difficult to implement). In these cases it can also be used as a reference algorithm for the generation of crop production functions with the highest potential yield, i.e., highest water use efficiency.

The NDP algorithm for closed-loop optimization has a wider range of application in irrigation operation. Once an approximate cost-to-go function is calculated it can be used for irrigation scheduling under any arbitrary management scheme. In the example application used in this paper the NDP algorithm showed its robustness in various runs. For example, only minor changes in yield occurred when $d_{\min}$ was varied between 0 and 10 days. The approximation of the cost-to-go function overcomes the 'curse of dimensionality' but still needs considerable time for the determination of the optimal weights of a linear basis function approximation of the cost-to-go by the policy iteration algorithm using LSTD. It is worthwhile to improve this method because closed-loop optimization offers some advantages over open-loop optimization including (1) feedback control which can respond immediately to external effects (e.g. rainfall), (2) stable performance even with model uncertainties or uncertainties of the initial or boundary conditions (e.g. climate conditions) and (3) reduced sensitivity to parameter variations.

Future work will focus on the application of both algorithms under uncertain climate conditions and/or soil hydraulic parameters. In this context, NDP overcomes the 'curse of modeling', which means that the transition probabilities do not have to be computed explicitly for stochastic DP. It uses the distribution of the random variables with no limitation placed on the stochastic model to simulate the system's behavior. Further investigations are under progress which already included more comprehensive irrigation models such as the FIM (Wöhling & Schmitz 2006a, b; Schmitz *et al.* 2007) and the SVAT models DAISY (Abrahamsen & Hansen 2000; Schütze & Schmitz 2010) and APSIM (Keating *et al.* 2003; Schütze *et al.* 2011) in the optimization of deficit irrigation systems.

Availability

The EA written in Matlab® is available on request from the first and second authors.

REFERENCES

Abrahamsen, P. & Hansen, S. 2000 DAISY: an open soil-crop-atmosphere system model. *Environ. Model. Software* **15** (3), 313–330.

Bellman, E. & Dreyfus, S. 1962 *Applied Dynamic Programming*. Princeton University Press, Princeton, NJ.

Bertsekas, D. P. 2000 *Dynamic Programming and Optimal Control*. Athena Scientific, Nashua, NH.

Boyan, J. A. 2002 Technical update: least-squares temporal difference learning. *Mach. Learn.* **49** (2–3), 233–246.

Bras, R. L. & Cordova, J. R. 1981 Intraseasonal water allocation in deficit irrigation. *Water Resour. Res.* **17** (4), 866–874.

Brown, P. D., Cochrane, T. A., Krom, T. D., Painter, D. J. & Bright, J. C. 2006 Optimal on-farm multicrop irrigation scheduling with limited water supply. In *Computers in Agriculture and Natural Resources. 4th World Congress Conference, Orlando, FL, USA* (F. Zazueta, J. Kin, S. Ninomiya & G. Schiefer, eds.). ASABE, St. Joseph, MI.

de Paly, M. & Zell, A. 2009 Comparison of efficiency and effectiveness of evolutionary algorithms for optimal irrigation scheduling. In *Proceedings of the12th European Conference on Genetic Programming, Lecture Notes in Computer Science 5481*, Springer, Berlin, pp. 159–170.

Doorenbos, J. & Kassam, A. 1979 *Yield Response to Water*. Technical Report, FAO, Irrigation and Drainage paper no. 33. FAO, Rome.

Gorantiwar, S. D., Smout, K. & Vairavamoorthy, K. 2006 Performance-based optimization of land and water resources within irrigation schemes. *I: Method. J. Irrigat. Drain Eng.* **132** (4), 332–340.

Gosavi, A. 2003 *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Kluwer, Dordrecht.

Hinnell, A. C., Lazarovitch, N., Furman, A., Poulton, M. & Warrick, A. W. 2010 Neuro-drip: estimation of subsurface wetting patterns for drip irrigation using neural networks. *Irrigat. Sci.* **28** (6), 535–544.

Jones, J. W., Hoogenboom, G., Porter, C. H., Boote, K. J., Batchelor, W. D., Hunt, L. A., Wilkens, P. W., Singh, U., Gijsman, A. J. & Ritchie, J. T. 2003 DSSAT cropping system model. *Eur. J. Agron.* **18**, 235–265.

Keating, B. A., Carberry, P. S., Hammer, G. L., Probert, M. E., Robertson, M. J., Holzworth, D., Huth, N. I., Hargreaves, J. N. G., Meinke, H., Hochman, Z., McLean, G., Verburg, K., Snow, V., Dimes, J. P., Silburn, M., Wang, E., Brown, S., Bristow, K. L., Asseng, S., Chapman, S., McCown, R. L., Freebairn, D. M. & Smith, C. J. 2003 An overview of APSIM, a model designed for farming systems simulation. *Eur. J. Agron.* **18**, 267–288.

Kholedian, M. R., Mailhol, J. C., Ruelle, P. & Rosique, P. 2009 Adapting PILOTE model for water and yield management under direct seeding system: the case of corn and durum wheat in a Mediterranean context. *Agr. Water. Manage.* **96**, 757–770.

Kumar, D. N., Raju, K. S. & Ashok, B. 2006 Optimal reservoir operation for irrigation of multiple crops using genetic algorithms. *J. Irrigat. Drain Eng.* **132** (2), 123–129.

Loganathan, G. & Elango, K. 2004 Revisiting optimal water-allocation under deficient supply. In: *Assessment and Management of Water Resources – AMWR 2004* (M. Kumar & M. Sekhar, eds.). Department of Civil Engineering, Indian Institute of Science, Bangalore, India.

Mailhol, J. 2005 Meteorological and hydraulic data from the Lavalette plot. Personal communication.

Montesinos, P., Camacho, E. & Alvarez, S. 2002 Application of generic algorithms for optimal seasonal furrow irrigation. *J. Hydroinf.* **4** (3), 145–156.

Nixon, J. B., Dandy, G. C. & Simpson, A. R. 2001 A genetic algorithm for optimizing off-farm irrigation scheduling. *J. Hydroinf.* **3** (1), 11–22.

Onwubolu, G. C. & Babu, B. V. 2004 *New Optimization Techniques in Engineering*. Springer, Berlin.

Prasad, A. S., Umamahesh, N. V. & Viswanath, G. K. 2006 Optimal irrigation planning under water scarcity. *J. Irrigat. Drain Eng*. **132** (3), 228–237.
Raghuwanshi, N. S. & Wallender, W. W. 1997 Economic optimization of furrow irrigation. *J. Irrigat. Drain Eng*. **5**, 377–385.
Rao, N. H., Sarma, P. B. S. & Chander, S. 1988 Irrigation scheduling under a limited water supply. *Agr. Water Manage*. **15**, 165–175.
Scheierling, S. M., Cardon, G. E. & Young, R. A. 1997 Impact of irrigation timing on simulated water crop production functions. *Irrigat. Sci*. **18** (1), 23–31.
Schmitz, G. H., Woehling, T., de Paly, M. & Schütze, N. 2007 GAIN-P: a new strategy to increase furrow irrigation efficiency. *Arab. J. Sci. Eng*. **32** (1C), 103–114.
Schütze, N. & Schmitz, G. H. 2010 OCCASION: new planning tool for optimal climate change adaption strategies in irrigation. *J. Irrigat. Drain Eng*. **136** (12), 836–846.
Schütze, N., Grundmann, J. & Schmitz, G. H. 2011 Prospects for integrated water resources management (IWRM) through the application of simulation-based optimization methods illustrated by the example of agricultural coastal arid regions in Oman. *Hydrologie und Wasserbewirtschaftung* **55**, 104–115.
Shang, S. & Mao, X. 2006 Application of a simulation based optimization model for winter wheat irrigation scheduling in north China. *Agr. Water. Manage*. **85** (3), 314–322.
Shang, S. H., Li, X. C., Mao, X. M. & Lei, Z. D. 2004 Simulation of water dynamics and irrigation scheduling for winter wheat and maize in seasonal frost areas. *Agr. Water Manage*. **68** (2), 117–133.
Shani, U., Tsur, Y. & Zemel, A. 2004 Optimal dynamic irrigation schemes. *Optim. Control Appl. Meth*. **25** (2), 91–106.
Singh, R. & Singh, J. 1997 Irrigation planning in wheat (triticum aestivum) under deep water table conditions through simulation modelling. *Agr. Water Manage*. **33** (1), 19–29.
Streichert, F. & Ulmer, H. 2005 *Javaeva – a Java Framework for Evolutionary Algorithms*. wsi-2005-06. Tech. rep., Center for Bioinformatics Tübingen, University of Tübingen.
Sunantara, J. D. & Ramirez, J. A. 1997 Optimal stochastic multicrop seasonal and intraseasonal irrigation control. *J. Irrigat. Drain Eng*. **123** (1), 39–48.
Sutton, R. S. & Barto, A. G. 1998 *Reinforcement Learning: An Introduction*. MIT Press, Boston, MA.
Wardlaw, R. & Bhaktikul, K. 2004 Application of genetic algorithms for irrigation water scheduling. *Irrigat. Drain* **53** (4), 397–414.
Wöhling, T. & Schmitz, G. H. 2006a Physically based seasonal modeling of the 1D surface – 2D subsurface flow in furrow irrigation systems I: model development. *J. Irrigat. Drain Eng*. **133** (6), 538–547.
Wöhling, T. & Schmitz, G. H. 2006b Physically based seasonal modeling of the 1D surface – 2D subsurface flow in furrow irrigation systems II: model test and evaluation. *J. Irrigat. Drain Eng*. **133** (6), 548–558.

First received 12 May 2010; accepted in revised form 24 November 2010. Available online 23 April 2011

APPENDIX

Algorithm 3 *tournament-selection* ($\mathbf{S}$, X^n, B_n, n_t)

Input: schedule $\mathbf{S}$ containing n_j irrigation events i at date d_i with volume v_i
current generation of schedules X^n
elitist set B_n

Ouput: new elitist set B_n, best individual schedule $\mathbf{S}'$

▷ choose $\mathbf{S}^{(1)} \dots \mathbf{S}^{(n_t-1)} \in X^n$

\> $\mathbf{S}' = \arg\max\,(\mathbf{Y}(\mathbf{S}), \mathbf{Y}(\mathbf{S}^{(1)}) \dots Y(\mathbf{S}^{(n_t-1)})$

if equal($\mathbf{S}'$, $\mathbf{S}$)

▷ $B_n = \{B_n, \mathrm{S}\}$

endif

return B_n, $\mathbf{S}'$

Algorithm 4 *crossover* ($\mathbf{S}$, $\mathbf{S}'$, p_t)

Input: schedule $\mathbf{S}$ containing n_j irrigation events i at date d_i with volume v_i
selected schedule $\mathbf{S}'$ containing n' irrigation events i at date d_i' with volume v_i' take over probability p_t

Ouput: new offspring $\mathbf{S}$

▷ $\mathbf{S}'' = \{\}$

for ($i = 1$; $i \le n_j$; i++)
  if ($p_t(\mathbf{s}_i)$)
    ▷ $\mathbf{S}'' = \{\mathbf{S}'', (d_i, v_i)\}$
  endif
endfor
for ($i = 1$; $i \le n'$; i++)
  if ($p_t(\mathbf{s}')$)
    ▷ $\mathbf{S}'' = \{\mathbf{S}'', (d_i', v_i')\}$
  endif
endfor
return $\mathbf{S}''$

Algorithm 5 *mutation* ($\mathbf{S}$, σ_d, σ_V)

Input: schedule $\mathbf{S}$ containing n_j irrigation events i at date d_i with volume v_i
mutation rates σ_d and σ_V for irrigation date d_i and volume v_i

Ouput: mutated schedule $\mathbf{S}$

for ($i = 1$; $i \le n_j$; i++)
  ▷ $d_i = d_i + \mathbf{randn}(0, \sigma_d)$
  ▷ $v_i = v_i + \mathbf{randn}(0, \sigma_V)$
endfor
return $\mathbf{S}$

Algorithm 6 *reconstruction* ($\mathbf{S}$, d_{min}, V_0)

Input: schedule $\mathbf{S}$ containing n_j irrigation events i at date d_i with volume v_i
minimal time between two irrigations d_{min} given water volume V_0

Ouput: reconstructed schedule $\mathbf{S}$

▷ end=1
while (end==1)
▷ end=0
  for ($i = 1$; $i \le (n_j - 1)$; i++)
    if (**abs** $(d_i - d_{i+1}) < d_{min}$)
      ▷ $d_i = \mathbf{min}(d_i, d_{i+1})$
      ▷ $v_i = v_i + v_{i+1}$
      ▷ delete $\mathbf{S}_{i+1}$
      ▷ $n_j = n_j - 1$
      ▷ end=1
    endif
  endfor
endwhile
for ($i = 1$; $i \le n_j$; i++)
  ▷ $v_i = \dfrac{v_i * \sum_{i=1}^{n_j} v_i}{V_0}$
endfor
return $\mathbf{S}$